

第 3 章



视频讲解

JavaScript 基础 ◀

建议学时：2

在前一章中学习了 HTML 语言,通过 HTML 可以利用标签描述一张网页,但是标签式的描述语言限制了网页在客户端进行的一些运算功能。本章学习 JavaScript 语言,JavaScript 嵌入 HTML 页面内,是一种运行在客户端并由浏览器进行解释执行的脚本语言,具有控制程序流程的功能。本章将学习其基本语法及基本对象。

3.1 JavaScript 简介

JavaScript 是一种网页脚本语言,虽然名字中含有 Java,但它与 Java 语言是两种完全不同的语言。不过,JavaScript 的语法和 Java 语言的语法非常类似。

JavaScript 代码可以很容易地嵌入 HTML 页面中,浏览器对 JavaScript 脚本程序进行解释执行。

3.1.1 第一个 JavaScript 程序

JavaScript 代码可以嵌入 HTML 中,它的基本写法如 firstPage.html 文件所示。

firstPage.html

```
<html>
  <body>
    <script type="text/javascript">
      window.alert("第一个 JavaScript 程序");    <!-- 弹出消息框 -->
    </script>
  </body>
</html>
```



图 3-1 页面运行效果

在保存为 HTML 页面后使用浏览器打开,将会弹出如图 3-1 所示的消息框。

注意：JavaScript 代码块“< script type=" text/javascript"> JavaScript 代码</script>”除了可以像上面一样写在< body></body>之间外,还可以写到< head></head>之间,其效果相同。

“< script type=" text/javascript"> JavaScript 代码</script>”也可以写为“< script language=" javascript"> JavaScript 代码</script>”。

JavaScript 与 Java 一样,对大小写是敏感的。

在 JavaScript 中注释有 3 种写法,一种是 HTML 注释的写法“<!--注释内容-->”,还有两种和 Java 语言相同,分别为“//单行注释”和“/* 多行注释 */”。

用户除了可以将 JavaScript 代码嵌入 HTML 中之外,还可以专门将 JavaScript 代码写在单独的文件中。

code.js

```
window.alert("第一个 JavaScript 程序");
```

然后在另外的 HTML 页面中插入以下代码来导入该文件。

```
<script src = "code.js" type = "text/javascript"></script>
```

此外,在 HTML 代码中可以写多个 JavaScript 代码块。

3.1.2 JavaScript 语法

1. 变量的定义

JavaScript 中的变量为弱变量类型,即变量的类型根据它被赋值的类型改变,定义一个变量使用的格式如下。

```
var 变量名
```

比如定义变量 arg 就可以使用“var arg”。如果将一个字符串赋给它,它就是 String 类型;如果将一个数组赋给它,它就是数组类型。

下面的 var.html 文件是一些变量定义的应用例子,代码如下。

var.html

```
<html>
<body>
  <script type = "text/javascript">
    var arg1,arg2,arg3;          <!-- 定义三个变量 -->
    var arg4 = 5;                <!-- 定义一个整型(Integer)变量 -->
    var arg5 = 10.0;             <!-- 定义一个浮点型(Float)变量 -->
    var arg6 = "你好!";          <!-- 定义字符型(String)变量 -->
    var arg7 = true;             <!-- 定义一个布尔类型(Boolean)变量 -->
    var arg8 = new Array("王","李","赵","张"); <!-- 定义字符串数组 -->
  </script>
</body>
</html>
```

需要注意的是,在 JavaScript 中变量未声明就使用是不会报错的,但很容易出现不可预知的错误,所以建议所有变量先声明后使用。

另外,函数 Number(字符串)可以将字符串转为数值;函数 String(数值)可以将数值转为字符串。

2. 函数的定义

在 JavaScript 中定义一个函数的基本格式如下。

```
function 函数名(参数列表){  
    return 值;  
}
```

用户也可以在使用中直接匿名定义,格式如下。

```
var arg1 = function(参数列表){  
    return 值;  
}
```

下面的 fun.html 文件是一个函数定义的应用示例,代码如下。

fun.html

```
<html>  
  <body>  
    <script type="text/javascript">  
      var arg0 = "欢迎使用 JavaScript";  
      print(arg0);  
      function print(arg1){  
        window.alert(arg1);  
      }  
    </script>  
  </body>  
</html>
```

其运行效果如图 3-2 所示。



图 3-2 页面运行效果

实际上,JavaScript 的语法和 Java 的语法基本类似,因此这里不作详细讲述。以上介绍的几个知识点是 JavaScript 与 Java 有差别的语法,其他的常用语句与 Java 类似,比如 if 判断语句,在 JavaScript 中的写法如下。

```
<html>  
  <body>  
    <script type="text/javascript">  
      var score = 67;  
      if(score >= 60){  
        window.alert("及格");  
      }  
    </script>  
  </body>  
</html>
```

```
        }else{
            window.alert("不及格");
        }
    }
</script>
</body>
</html>
```

又如 for 循环的写法如下。

```
<html>
<body>
    <script type="text/javascript">
        for(var i=1;i<10;i++){
            window.alert(i);
        }
    </script>
</body>
</html>
```

以上写法是与 Java 一样的。

下面用循环举一个实际的例子。编写一个恶意程序,用户打开,会不断弹出消息框。其代码如“恶意网页.html”文件所示。

恶意网页.html

```
<html>
<body>
    <script language="javascript">
        str=new Array("你受骗了","你真的受骗了","真笨啊");
        while(true){
            for(i=0;i<str.length;i++){
                window.alert(str[i]);
            }
        }
    </script>
</body>
</html>
```

该代码运用了 JavaScript 中的循环,使得消息框怎么点都不会结束,而且无法关掉浏览器,只能通过关闭进程结束。读者可以进行实验。

3.2 JavaScript 内置对象

除了可以在代码里面进行简单的编程之外,还可以通过 JavaScript 提供的内置对象对网页进行操作,内置对象由浏览器提供,可以直接使用,不用事先定义。例如 3.1.1 节例子中的“window.alert(“第一个 JavaScript 程序”)”,其中的 window 就是一个内置对象。

使用最多的内置对象有以下 4 个,并且本书之后的学习将主要围绕这 4 个对象展开。

- (1) window: 负责操作浏览器窗口,负责窗口的状态、开/闭等。
- (2) history: 可以代替后退(前进)按钮访问历史记录,从属于 window。
- (3) document: 负责操作浏览器载入的文档(HTML 文件),从属于 window。
- (4) location: 访问地址栏,也从属于 window。

注意: 如果一个对象从属于另一个对象,在使用时用“.”隔开,例如 window.document.XXX,但是如果从属于 window 对象,window 可以省略,例如 window.document.XXX 可以写为 document.XXX。

3.2.1 window 对象

window 对象的作用如下。

1. 出现提示框

window 对象可以跳出提示框,主要有如下功能。

- window.alert("内容"): 出现消息框。
- window.confirm("内容"): 出现确认框。
- window.prompt("内容"): 出现输入框。

下面 window1.html 文件中代码的功能是显示一些提示框。

window1.html

```
<html>
  <body>
    <script type="text/javascript">
      //1:消息框
      window.alert("消息框");
      //2:确认框,根据 result 的值 true 或者 false 来判断
      result = window.confirm("您确认提交吗?");
      //3: 输入框,str 为输入的值,如果取消,str 的值为 null
      str = window.prompt("请您输入一个字符串","");
    </script>
  </body>
</html>
```

用浏览器打开该文件将会依次出现如图 3-3 所示的提示框。



图 3-3 提示框运行效果

浏览器弹出提示框后载入页面将会停滞,直到用户做出操作动作,其中消息框的实际运用最为广泛,确认框其次,输入框则较为少见。

2. 打开、关闭窗口

window 对象还用于控制窗口的状态和开/闭。打开窗口主要使用 window 对象的 open() 函数。

下面的 window2.html 文件是一个打开窗口的应用实例,代码如下。

window2.html

```
<html>
  <body>
    <script type="text/javascript">
      window.status = "出现新窗口";
      //打开新窗口
      newWindow = window.open("window1.html", "new1",
        "width=300,height=300,top=500,left=500");
      //可以通过返回值来控制新窗口
      //newWindow.close();      //关闭窗口
    </script>
  </body>
</html>
```

在本例中首先让窗口的状态栏显示字符串“出现新窗口”,然后打开一个新窗口 window1.html,命名为 new1,并指定宽度、高度和其位置。运行效果如图 3-4 所示。

在源程序中,“newWindow.close();”表示关闭 newWindow。

window 对象的 status 属性值将显示在浏览器左下角的状态栏中,如图 3-5 所示。



图 3-4 运行效果



图 3-5 项目运行效果

综上所述,window.open()在网页制作中的使用非常广泛,参数有 3 个,第 1 个是新窗口的地址,第 2 个是新窗口的名称,第 3 个是新窗口的状态,其中新窗口状态可设置如下属性。

- toolbar: 是否有工具栏,可选值为 1 和 0。
- location: 是否有地址栏,可选值为 1 和 0。
- status: 是否有状态栏,可选值为 1 和 0。
- menubar: 是否有菜单栏,可选值为 1 和 0。
- scrollbars: 是否有滚动条,可选值为 1 和 0。
- resizable: 是否能改变大小,可选值为 1 和 0。
- width 和 height: 窗口的宽度和高度,用像素表示。
- left 和 top: 窗口左上角相对于桌面左上角的 x 和 y 坐标。

各属性值用逗号隔开,如下面的代码所示。

```
newWindow = window.open("window1.html", "new1",
  "toolbar=0,width=300,height=300,top=500,left=500");
```

3. 定时器

window 对象负责管理和控制页面的定时器,定时器的作用是让某个函数隔一段时间之后运行一次,格式如下。

```
timer = window.setTimeout("需要运行的函数","时间(用毫秒计)");
```

如果要清除定时器,可以用如下代码。

```
clearTimeout(timer);
```

下面的 timer.html 是一个定时器的应用实例,代码如下。

timer.html

```
<html>
  <body>
    <script type="text/javascript">
      //setTimeout 让函数在某段时间之后运行 1 次,参数 2 是毫秒数
      timer = window.setTimeout("fun1()", "1000");
      var i = 0;
      function fun1(){
        i++;
        window.status = i;
        if(i == 100){
          window.clearTimeout(timer);      //清除定时器,否则会一直运行
          return;
        }
        timer = window.setTimeout("fun1()", "1000");
      }
    </script>
  </body>
</html>
```

其运行效果如图 3-6 所示。

这样,每隔 1 秒钟状态栏中的数字将会加 1,直到 100,之后将一直持续 100 的状态,不再改变。



图 3-6 定时器运行效果

设置定时器可以使网页定时刷新,这在一些要求计时功能的网页中经常被用到。

3.2.2 history 对象

history 对象包含用户的浏览历史等信息,使用这个对象是因为它可以代替后退(前进)按钮访问历史记录,该对象从属于 window。

history 对象最常用的函数如下。

- history.back(): 返回上一页,相当于单击浏览器上的后退按钮。
- history.forward(): 返回下一页,相当于单击浏览器上的前进按钮。
- window.history.go(n): n 为整数,正数表示向前进 n 格页面,负数表示向后退 n 格页面。

下面的 history.html 文件是 history 对象的应用实例,代码如下。

history. html

```
<html>
  <body>
    <a onclick = "history.forward()">前进</a>
    <a onclick = "history.back()">后退</a>
  </body>
</html>
```

运行 history. html,效果如图 3-7 所示。

单击“前进”或者“后退”,其效果和单击浏览器上的按钮一样。

前进 后退

注意: 此处用到了网页元素的事件,由于篇幅所限,本章仅仅用到单击事件(onclick),对于其他事件,读者可以参考相应文档。

图 3-7 history. html
运行效果

3.2.3 document 对象

document 对象从属于 window,其功能如下。

1. 在网页上输出

在网页输出方面,最常见的函数是 writeln()。

下面的 document1. html 文件是 writeln()函数的应用实例,代码如下。

document1. html

```
<html>
  <body>
    <script type = "text/javascript">
      document.writeln("你好");
    </script>
  </body>
</html>
```

你好

其运行效果如图 3-8 所示。

图 3-8 document1. html
运行效果

writeln()函数为输出一些简单却重复的代码提供了很大的便利,在下面的例子中将要使用表格显示一个 8×8 的国际象棋棋盘,正常的方法需要写一个 8 行 8 列表格的代码,但会使源代码非常冗长。下面的 chess. html 文件是使用 writeln()函数实现的方法。

chess. html

```
<html>
  <body>
    <script type = "text/javascript">
      document.writeln("<table width = 400 height = 400 border = 1>");
      for(i = 1;i <= 8;i++){
        document.writeln("<tr>");
        for(j = 1;j <= 8;j++){
          color = "black";
          if((i + j) % 2 == 0){
```

```

        color = "white";
    }
    document.writeln("<td bgcolor = " + color + "></td>");
}
document.writeln("</tr>");
}
document.writeln("</table>");
</script>
</body>
</html>

```

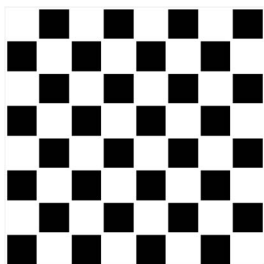


图 3-9 棋盘运行效果

借助 `writeln()` 和循环方法省去了很多 HTML 代码的编写。该例的运行效果如图 3-9 所示。

2. 设置网页的属性

使用 `document` 对象可以进行一些简单网页属性的设置,例如网页的标题、颜色等,并且可以得到网页的某些属性,例如当前地址。其比较常用的设置包括通过 `document.title` 来访问标题,通过 `document.location` 来获取当前网页的地址等。

下面的 `document2.html` 文件是一个设置网页属性的应用实例,代码如下。

document2.html

```

<html>
<body>
<script type="text/javascript">
    function fun(){
        document.title="新的标题";        //设置网页标题
        window.alert(document.location);    //得到当前网页的地址
    }
</script>
<input type="button" onclick="fun()" value="按钮">
</body>
</html>

```

运行后,单击“按钮”,将会弹出一个消息框,内容为当前页面的地址,并且网页的标题将改变为“新的标题”。对于其他功能,读者可以参考相应文档。

3. 访问文档元素,特别是表单元素

使用 `document` 对象可以访问文档中的元素(例如图片、表单、表单中的控件等),前提是元素的 `name` 属性是确定的。其访问方法如下。

```
document.元素名.子元素名...
```

比如名为 `form1` 的表单中有一个文本框 `account`,其中的内容可以用如下代码获得。

```
var account = document.form1.account.value;
```

下面的 `document3.html` 文件是访问表单元素的例子,其中有两个文本框、一个按钮,

输入两个数字,单击“求和”按钮,将显示两个数字的和。其代码如下。

document3. html

```
<html>
<body>
  <script type="text/javascript">
    function add(){
      //得到这两个文本框的内容
      n1 = Number(document.form1.txt1.value);
      n2 = Number(document.form1.txt2.value);
      document.form1.txt3.value = n1 + n2;
    }
  </script>
  <form name="form1">
    <input name="txt1" type="text"><br>
    <input name="txt2" type="text"><br>
    <input type="button" onclick="add()" value="求和"><br>
    <input name="txt3" type="text"><br>
  </form>
</body>
</html>
```

运行后文本框为空,在第1行和第2行文本框中填入数字并单击“求和”按钮,运行效果如图3-10所示。

由于使用 document 对象可以得到网页中元素的值,所以它在客户端的验证中用得非常广泛,比如在注册或登录中可以使用 JavaScript 得到表单中的值,然后通过判断做出相应的反应。

下面的 validate. html 文件是一个利用 JavaScript 判断表单中值的实例,代码如下。

validate. html

```
<html>
<body>
  <script type="text/javascript">
    function validate(){
      //得到这两个文本框的内容
      account = document.loginForm.account.value;
      password = document.loginForm.password.value;
      if(account == ""){
        alert("账号不能为空");
        document.loginForm.account.focus();      //聚焦
        return;
      }
      else if(password == ""){
        alert("密码不能为空");
        document.loginForm.password.focus();
        return;
      }
      document.loginForm.submit();
    }
  </script>
  欢迎您登录:
```

图3-10 求和运行效果

```

< form name = "loginForm">
    输入账号:< input name = "account" type = "text">< br >
    输入密码:< input name = "password" type = "password">< br >
    < input type = "button" onclick = "validate()" value = "登录">
</form>
</body>
</html>

```

欢迎您登录:



图 3-11 验证效果

特别提醒: “document.loginForm.account.focus();”为聚焦函数,是使光标移动到调用这个函数的元素位置;“document.loginForm.submit();”为提交表单,与单击“登录”按钮的效果一样。

这样进行验证可以减少服务器遭到恶意登录的可能。

运行网页,不输入账号就进行登录,效果如图 3-11 所示。

从上面的程序可以看出,当用户没有输入账号或密码就单击“登录”按钮时将弹出提示填写账号或密码的消息框,直到都填写完整,表单才能提交。

3.2.4 location 对象

location 对象可以访问浏览器的地址栏,它也从属于 window,其最常见的功能就是跳转到另一个网页,跳转的方法是修改 location 对象的 href 属性。

下面的 location1.html 文件是一个网页跳转的应用实例,代码如下。

location1.html

```

< html >
  < body >
    < script type = "text/javascript">
      function locationTest(){
        window.location.href = "image.jpg";
      }
    </script>
    < input type = "button" onclick = "locationTest()" value = "按钮">
    < a href = "image.jpg">到图片</a>
  </body>
</html>

```

其运行效果如图 3-12 所示。

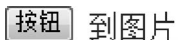


图 3-12 location1.html 运行效果

单击“按钮”和单击“链接”的效果是一样的,都会跳转到如图 3-13 所示的页面。

比较常见的另一个应用是定时跳转,在使用时可以结合 window 的定时器。下面的 location2.html 文件是它的具体实现,代码如下。



图 3-13 跳转到目标页面

location2. html

```
<html>
  <body>
    欢迎您登录,3 秒钟转到首页.....
    <script type="text/javascript">
      window.setTimeout("toIndex()", "3000");    //在 3 秒钟后运行一次 toIndex()
      function toIndex(){
        window.location.href = "image. jpg";
      }
    </script>
  </body>
</html>
```

其运行效果如图 3-14 所示。

欢迎您登录, 3秒钟转到首页.....

图 3-14 location2. html 运行效果

3 秒钟后界面效果如图 3-13 所示。

本章小结

本章学习了 JavaScript 语言的基本语法和基本内置对象,并通过一些常见应用讲解了这些知识点的使用方法。

值得一提的是,本章只讲解了 JavaScript 的基本内容,如果读者想要向客户端编程方面发展,需要了解更多的 JavaScript 知识。

课后习题

一、填空题

1. 浏览器对 JavaScript 脚本程序进行_____执行。
2. JavaScript 的 3 种注释写法为_____、_____、_____。
3. 若将 JavaScript 代码写在单独的 test.js 文件中,需要在调用它的 HTML 页面中插

入的代码是_____。

4. JavaScript 的内置对象_____负责操作浏览器窗口,其中_____方法可以弹出消息框,_____方法可以关闭窗口,_____方法可以打开新窗口。

5. _____对象包含用户的浏览历史等信息,其中_____方法相当于单击浏览器上的后退按钮,_____方法相当于单击浏览器上的前进按钮。

6. 在名为 form 的表单中有一个文本框 account,其中的内容可以用代码_____获得。

7. 用 location 对象实现跳转到网页 a.html 的代码是_____。

8. document 对象从属于_____对象。

9. 用 document 对象可以进行一些简单网页属性的设置,通过_____来访问标题,通过_____来获取当前网页的地址。

二、选择题

1. 下列关于 JavaScript 的说法错误的是()。

- A. JavaScript 的语法和 Java 语言的语法非常类似
- B. JavaScript 中的变量是弱变量类型,即变量的类型根据它被赋值的类型改变
- C. JavaScript 对大小写是敏感的
- D. 服务器对 JavaScript 脚本程序进行编译、运行

2. 在 HTML 页面上编写 JavaScript 代码时应编写在()标签之间。

- A. < body ></body >
- B. < javascript ></javascript >
- C. < script ></script >
- D. < form ></form >

3. 在下面的 JavaScript 语句中,()定义了一个整型变量并赋值为 10。

- A. var if=10
- B. var larg=10
- C. var arg1=10.0
- D. var arg1=10

4. window.setTimeout("fun()",1000)表示的意思是()。

- A. 间隔 1 秒后,fun()函数被调用 1 次
- B. 间隔 1000 秒后,fun()函数被调用 1 次
- C. 间隔 1 秒后,fun()函数被调用 1000 次
- D. 间隔 1 毫秒后,fun()函数被调用 1000 次

5. window 对象的()属性用来指定浏览器状态栏中显示的临时消息。

- A. title
- B. status
- C. toolbar
- D. location

6. 在 history 对象中不能实现网页前进效果的方法是()。

- A. forward()
- B. back()
- C. go(1)
- D. go(2)

7. 在浏览器的状态栏中显示“这是状态栏”消息的代码是()。

- A. window.status="这是状态栏"
- B. window.status("这是状态栏")
- C. status("这是状态栏")
- D. status("这是状态栏","")

8. 下列打开新窗口的代码中正确的是()。

- A. `window.open("window2.html", "new")`
- B. `window.open("window2.html", "new", "")`
- C. `window.open("window2.html")`
- D. `window.open("new", "window2.html")`

9. 在代码`< body onLoad="f1()" onError="f2()">`

`< input onFocus="g1()" onClick="g2()">`

`</body>`中,一定会被调用的方法是()。

- A. `f1()`
- B. `f2()`
- C. `g1()`
- D. `g2()`

三、上机习题

1. 编写一个金额找零的系统,用输入框输入一个整数,表示找零的数量,数值为1~100。假如系统中有50、20、10、5、1这5种面额的纸币,显示每种纸币应该找的数量。例如78元应该为50元1张、20元1张、5元1张、1元3张。
2. 在表单中输入5本书的价格,显示这5本书价格的和。
3. 用 `document` 对象在屏幕上打印100个“欢迎”。
4. 用表单输入10本书的价格,然后显示这10本书的最高价格、最低价格和平均价格。